



# Arquitetura de Plataforma de Dados e Modern Data Stack

## Padrões de Arquitetura Monolito, Microserviços e Event-Driven



Padrões de arquitetura definem as práticas recomendadas para construir sistemas de software. Eles descrevem as relações entre componentes, os padrões de comunicação e a organização global da estrutura do sistema. Vamos explorar os padrões de Monolito, Microserviços e Event-Driven:

### Monolito

**Definição:** Em uma arquitetura monolítica, todos os componentes do software (interface do usuário, lógica de negócios, manipulação de dados, etc.) são interligados e interdependentes, sendo combinados em uma única base de código e executados como uma única unidade.

#### **Características:**

- **Simplicidade:** Mais fácil de desenvolver, testar e depurar, especialmente em fases iniciais.
- **Deploy Único:** A aplicação é lançada como uma única unidade, o que pode tornar os processos de deploy e escalabilidade mais simples em certos cenários.
- **Acoplamento:** Alterações em uma parte do sistema podem afetar outras, o que pode tornar a manutenção e a introdução de novos recursos mais desafiadoras à medida que a base de código cresce.

**Desvantagens:** Dificuldades de escalabilidade, maiores tempos de deploy, risco de falhas que afetam o sistema como um todo e potenciais desafios de modernização.

### Microserviços

**Definição:** Na arquitetura de microserviços, uma aplicação é construída como uma coleção de serviços menores e independentes, cada um executando uma funcionalidade específica. Cada serviço é autônomo e comunica-se com outros através de APIs.

#### **Características:**

- **Independência:** Cada serviço pode ser desenvolvido, testado, implantado e escalado de forma independente.
- **Poliglotismo:** Diferentes microserviços podem usar diferentes linguagens de programação, bancos de dados ou ferramentas, conforme a necessidade.
- **Resiliência:** Falhas em um microserviço podem não afetar diretamente outros.

**Desvantagens:** Complexidade aumentada em termos de gerenciamento de serviços, necessidade de ferramentas para coordenação e monitoramento, desafios em garantir transações atômicas entre serviços.

### Event-Driven (Baseada em Eventos)

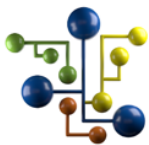
**Definição:** Em uma arquitetura baseada em eventos, os componentes produzem ou consomem eventos. Estes componentes reagem aos eventos e tomam decisões baseadas neles.

**Características:**

- Assincronicidade: Os componentes não precisam esperar uma resposta imediata após enviar um evento, permitindo operações não bloqueantes.
- Escalabilidade: Como os componentes são geralmente desacoplados, eles podem ser escalados de forma independente.
- Resiliência: A falha de um componente pode não afetar outros diretamente.

**Desvantagens:** Pode ser desafiador garantir a consistência dos dados, a ordem dos eventos pode ser crucial e pode ser difícil de gerenciar, e pode exigir um redesign significativo para sistemas que originalmente não foram construídos de forma orientada a eventos.

Cada um desses padrões tem seus próprios benefícios e desafios e a escolha de qual usar depende das necessidades e contextos específicos do projeto ou organização.



**Equipe DSA**

**Muito Obrigado!**  
**Continue Trilhando Uma Excelente Jornada de Aprendizagem.**